

CLAIMS

What is claimed is:

1. A distributed processing computer apparatus for use in systems, the apparatus comprising:

a plurality of processes executing on at least one processor;

at least one application executing in a pure distributed mode where said application is distributed in an active condition among more than one of said processes on said processors;

a system controller for controlling system activation and initial load distribution;

a router for providing communications between at least one said application and other applications independent of application locations;

an ADSM for providing distributed functionality in said application; and

an ALDM for distributing incoming events to said application.

2. The computer apparatus recited in claim 1 wherein said system controller also provides procedures for controlling any one or more members of the group consisting of fault recovery, load redistribution, system topology, and system maintenance.

3. The computer apparatus recited in claim 1 further comprising a plurality of resource sets each being a unit of distribution, and said application using more than one said resource set.

4. The computer apparatus recited in claim 3 wherein shared data in said application is modified by a master critical resource set and updated onto shadow resource sets on all copies of said application and private data in said application is modified by active non-critical resource sets.

5. The computer apparatus recited in claim 3 wherein said ADSM provides API for making a resource set active.

6. The computer apparatus recited in claim 3 wherein said ADSM provides API for making a resource set standby and to warm start said standby resource set.

7. The computer apparatus recited in claim 3 wherein said ADSM provides API for making a resource set out of service.

8. The computer apparatus recited in claim 3 wherein said ADSM provides API to disable peer update towards a resource set.

9. The computer apparatus recited in claim 4 wherein said ALDM distributes the processing load by mapping incoming events to said resource sets and sending events to said active resource set.

10. The computer apparatus recited in claim 3 wherein said ALDM provides API to set the weight of a resource set.

11. The computer apparatus recited in claim 1 further comprising a load manager for providing dynamic load balancing for said applications by using APIs selected from the group consisting of:

APIs of said ALDM,
APIs of said ADSM,
APIs of said router, and
APIs of said system controller.

12. The computer apparatus recited in claim 4 wherein said router provides API to send messages to said active resource set of said application.

13. The computer apparatus recited in claim 4 wherein said router provides API to set and clear active mapping for said resource sets.

14. The computer apparatus recited in claim 4 wherein said router provides API to set and clear standby mapping for said resource sets.

15. The computer apparatus recited in claim 4 wherein said router provides API to set and clear master mapping for said master critical resource set and to add and remove shadow mapping from a multicast list for said critical resource set.

16. The computer apparatus recited in claim 3 wherein said router provides API to hold and release messages for said resource sets.

17. The computer apparatus recited in claim 3 wherein said router provides API to perform adjacent ping for flushing communication channels and to peersync messages held for said resource sets with said router.

18. The computer apparatus recited in claim 3 wherein said router provides API to send update messages to a standby resource set.

19. The computer apparatus recited in claim 4 wherein said router provides API to send messages to all said shadows in a multicast list of said critical resource set.

20. The computer apparatus recited in claim 4 wherein said system controller is configured with all of the said applications in the system, with mode of operation for each said application, said critical and non-critical resource sets information of each said application and service user/provider relationship between said applications.

21. The computer apparatus recited in claim 3 wherein said system controller provides resource set level API to make a resource set active.

22. The computer apparatus recited in claim 3 wherein said system controller provides resource set level API to make a resource set standby.

23. The computer apparatus recited in claim 3 wherein said system controller provides resource set level API to make a resource set out of service.

24. The computer apparatus recited in claim 3 wherein said system controller provides resource set level API to perform any one or more of the group consisting of forced switchover, controlled switchover, forced move and controlled move operation.

25. The computer apparatus recited in claim 3 wherein said system controller provides application level enable node API to introduce a process with at least one application into a system during initialization, for scaling an operational system, and wherein said system controller implements algorithms to redistribute the load between all said processes by movement of resource sets.

26. The computer apparatus recited in claim 3 wherein said system controller provides application level disable node API to

recover from the failure of at least one application in a process and wherein said system controller redistributes the load by movement of resource sets.

27. The computer apparatus recited in claim 3 wherein said system controller provides application level disable node API to shutdown at least one said application in a process and wherein said system controller redistributes the load by movement of resource sets.

28. A fault tolerant computer apparatus for use in systems, the apparatus comprising:

a plurality of processes executing on at least one processor;

at least one application executing in a pure fault tolerant mode where said application is in an active condition on one said process and in a standby condition on another said process on said processors;

a system controller for controlling system activation and failure recovery;

a router for providing communications between at least one said application and other applications independent of application locations; and

an ADSM for providing fault tolerant functionality in said application and wherein said application is represented by a single resource set.

29. The computer apparatus recited in claim 28 wherein data in said application is modified by a single active resource set and updated on a standby resource set.

30. The computer apparatus recited in claim 28 wherein said ADSM provides API for making said single resource set active.

31. The computer apparatus recited in claim 28 wherein said ADSM provides API for making said single resource set standby and to warm start said standby resource set.

32. The computer apparatus recited in claim 28 wherein said ADSM provides API for making said single resource set out of service.

33. The computer apparatus recited in claim 28 wherein said ADSM provides API to disable peer update towards said single resource set.

34. The computer apparatus recited in claim 29 wherein said router provides API to send messages to said active resource set of said application

35. The computer apparatus recited in claim 29 wherein said router provides API to set and clear active mapping for said single resource set.

36. The computer apparatus recited in claim 29 wherein said router provides API to set and clear standby mapping for said single resource set.

37. The computer apparatus recited in claim 28 wherein said router provides API to hold and release messages for said single resource set.

38. The computer apparatus recited in claim 28 wherein said router provides API to perform adjacent ping for flushing communication channels and to peersync messages held for said resource set with said Router.

39. The computer apparatus recited in claim 29 wherein said router provides API to send update messages to said standby resource set.

40. The computer apparatus recited in claim 28 wherein said system controller is configured with all of the said applications in the system, with mode of operation for each said application, and service user/provider relationship between said applications.

41. The computer apparatus recited in claim 28 wherein said system controller provides resource set level API to make said resource set active.

42. The computer apparatus recited in claim 28 wherein said system controller provides resource set level API to make said resource set standby.

43. The computer apparatus recited in claim 28 wherein said system controller provides resource set level API to make said resource set out of service.

44. The computer apparatus recited in claim 28 wherein said system controller provides resource set level API to perform either one of the group consisting of forced switchover operation and controlled switchover operation.

45. The computer apparatus recited in claim 28 wherein said system controller provides application level enable node API to introduce a process with at least one application into a system during initialization.

46. The computer apparatus recited in claim 28 wherein said system controller provides application level disable node API to recover from the failure of at least one said application in one of said processes.

47. The computer apparatus recited in claim 28 wherein said system controller provides application level disable node API to shutdown at least one said application in one of said processes.

48. A distributed processing, fault tolerant computer apparatus for use in systems, the apparatus comprising:

a plurality of processes executing on at least one processor;

at least one application executing in a distributed fault tolerant mode where said application is in an active condition on more than one of said processes and is in a standby condition on at least one of said processes on said processors;

a system controller for controlling system activation, failure recovery and initial load distribution;

a router for providing communications between at least one said application and other applications independent of application locations;

an ADSM for providing distributed fault tolerant functionality in said application; and

an ALDM for distributing incoming events to said application.

49. The computer apparatus recited in claim 48 wherein said system controller also provides procedures for controlling any one or more members of the group consisting of load redistribution, system topology and system maintenance.

50. The computer apparatus recited in claim 48 further comprising a plurality of resource sets each being a unit of distribution and wherein said application uses more than one said resource set.

51. The computer apparatus recited in claim 50 wherein shared data in said application is modified by a master critical resource set and updated onto shadow resource sets on all copies of said application and private data in said application is modified by active non-critical resource sets and updated onto standby resource sets.

52. The computer apparatus recited in claim 50 wherein said ADSM provides API for making a resource set active.

53. The computer apparatus recited in claim 50 wherein said ADSM provides API for making a resource set standby and to warm start said standby resource set.

54. The computer apparatus recited in claim 50 wherein said ADSM provides API for making a resource set out of service.

55. The computer apparatus recited in claim 50 wherein said ADSM provides API to disable peer update towards a resource set.

56. The computer apparatus recited in claim 51 wherein said ALDM distributes the processing load by mapping incoming events to said resource sets and sending events to an active resource set.

57. The computer apparatus recited in claim 50 wherein said ALDM provides API to set the weight of the said resource sets.

58. The computer apparatus recited in claim 48 further comprising a load manager for providing dynamic load balancing for said applications by using APIs selected from the group consisting of:

APIs of said ALDM,
APIs of said ADSM,
APIs of said router, and
APIs of said system controller.

59. The computer apparatus recited in claim 51 wherein said router provides API to send messages to said active resource sets of said application.

60. The computer apparatus recited in claim 51 wherein said router provides API to set and clear active mapping for said resource sets.

61. The computer apparatus recited in claim 51 wherein said router provides API to set and clear standby mapping for said resource sets.

62. The computer apparatus recited in claim 51 wherein said router provides API to set and clear master mapping for said critical master resource set and to add and remove shadow mapping from a multicast list for said critical resource sets.

63. The computer apparatus recited in claim 50 wherein said router provides API to hold and release messages for said resource sets.

64. The computer apparatus recited in claim 50 wherein said router provides API to perform adjacent ping for flushing communication channels and to peersync messages held for said resource sets with said router.

65. The computer apparatus recited in claim 51 wherein said router provides API to send update messages to said standby resource sets.

66. The computer apparatus recited in claim 51 wherein said router provides API to send messages to all said shadows in the multicast list of said critical resource set.

67. The computer apparatus recited in claim 51 wherein said system controller is configured with all of the said applications in a system, with mode of operation for each said application, with said critical and non-critical resource sets information of each said application and service user/provider relationship between said applications.

68. The computer apparatus recited in claim 50 wherein said system controller provides resource set level API to make a resource set active.

69. The computer apparatus recited in claim 50 wherein said system controller provides resource set level API to make a resource set standby.

70. The computer apparatus recited in claim 50 wherein said system controller provides resource set level API to make a resource set out of service.

71. The computer apparatus recited in claim 50 wherein said system controller provides resource set level API to perform one or more of the group consisting of forced switchover, controlled switchover, forced move and controlled move operation.

72. The computer apparatus recited in claim 50 wherein said system controller provides application level enable node API to introduce a process with at least one application into a system

during initialization, for scaling an operational system, and wherein said system controller implements algorithms to redistribute the load between all said processes by movement of resource sets.

73. The computer apparatus recited in claim 50 wherein said system controller provides application level disable node API to recover from the failure of at least one application in a process and wherein said system controller redistributes the load by movement of resource sets.

74. The computer apparatus recited in claim 50 wherein said system controller provides application level disable node API to shutdown at least one application in a process and wherein said system controller redistributes the load by movement of resource sets.

75. A distributed processing, computer apparatus for use in systems, the apparatus comprising:

a plurality of processes executing on at least one processor;

at least one application executing in a pure distributed mode where said application is distributed in an active condition among more than one of said processes on said processors;

a system controller for controlling system activation and initial load distribution;

a router for providing communications between at least one said application and other applications independent of

application locations;

an update module for providing distributed functionality in said application; and

a load distributor for distributing incoming events to said application.

76. A fault tolerant computer apparatus for use in systems, the apparatus comprising:

a plurality of processes executing on at least one processor;

at least one application executing in a pure fault tolerant mode where said application is in an active condition on one said process and in a standby condition on another said process on said processors;

a system controller for controlling system activation and failure recovery;

a router for providing communications between at least one said application and other applications independent of application locations; and

an update module for providing fault tolerant functionality in said application and wherein said application is represented by a single reserved resource set.

77. A distributed processing, fault tolerant computer apparatus for use in systems, the apparatus comprising:

a plurality of processes executing at least one processor;

at least one application executing in a

distributed fault tolerant mode where said application is in an active condition on more than one of said processes and is in a standby condition on at least one of said processes on said processors;

a system controller for controlling system activation, failure recovery and initial load distribution;

a router for providing communications between at least one said application and other applications independent of application locations;

an update module for providing distributed fault tolerant functionality in said application; and

a load distributor for distributing incoming events to said application.

78. A fault tolerant, distributed processing, computer apparatus for use in systems, the apparatus comprising:

a plurality of processes, executing on at least one processor;

said processes executing an application in the same mode as at least one other application or in a mode different from said one other application, said same and different modes being:

a) a pure distributed mode where an application is distributed among said processes in an active condition;

b) a pure fault-tolerant mode where an application executes in at least one process in an active condition and in at least one process in a standby condition; and

c) a distributed fault-tolerant mode where an application is distributed on multiple processes in an active condition and on at least one process in a standby condition.

79. A method in a computer apparatus for fault tolerant and distributed processing of at least one application in a plurality of processes running on at least one processor, the method comprising the steps of:

executing said application in a distributed fault tolerant mode wherein said application is distributed in an active condition among more than one process and is in standby condition on at least one said process on said processors;

providing a plurality of resource sets as units of distribution of said application; and

a master critical resource set modifying shared data in said application and updating to a shadow resource set of said application on said processes and an active non-critical resource set modifying private data in said application and updating to a standby resource set of said application on another said process.

80. The method recited in claim 79, further comprising the steps of:

bringing said resource sets into either of active or standby state on said processes; and

said active resource set processing input events and sending update information to said standby resource set.

81. The method recited in claim 79, comprising the further step of using a warmstart procedure to bring said resource sets into standby state from out of service state.

82. The method recited in claim 79, comprising the further step of distributing the processing load of said application by mapping incoming events to said resource sets of said application and sending events to active resource sets.

83. The method recited in claim 79, further comprising the step of providing communication between said application and other applications independent of application location and carrying out said communication external to the application by routing an event to the process where a mapped resource set is active.

84. The method recited in claim 80, further comprising the step of transparently sending update messages from said active resource set to a corresponding said standby resource set by performing routing external to said application and routing messages to the process where the resource set is standby.

85. The method recited in claim 80, further comprising the steps of bringing said standby resource sets into the active state for recovering from a failure of active resource sets and routing events to new active resource sets.

86. The method recited in claim 79, further comprising the step of dynamic load balancing by either moving the resource sets from one said process to other said process or by mapping new events to relatively idle resource sets.

87. A method in a computer apparatus for distributed processing of at least one application in a plurality of processes running on at least one processor; the method comprising the steps of:

executing said application in a pure distributed mode wherein said application is distributed in an active condition among more than one process;

providing a plurality of resource sets as units of distribution of said application;

a master critical resource set modifying shared data in said application and updating to a shadow resource set of said application on said processes and an active non-critical resource set modifying private data in said application.

88. The method recited in claim 87, further comprising the steps of:

bringing said resource sets into active state on said processes; and

said active resource set processing input events.

89. The method recited in claim 87, further comprising the step of using a warmstart procedure to bring said resource sets into shadow state from out of service state.

90. The method recited in claim 87, further comprising the step of distributing the processing load of said application by mapping incoming events to said resource sets of said application and sending events to active resource sets.

91. The method recited in claim 87, further comprising the step of providing communication between said application and other applications independent of application location and carrying out said communication external to the application by routing an event to the process where a mapped resource set is active.

92. The method recited in claim 87, further comprising the step of transparently sending update messages from said active resource set to a corresponding said shadow resource sets by performing routing external to said application and routing messages to the processes where a resource set is shadow.

93. The method recited in claim 87, further comprising the step of dynamic load balancing by either moving the resource sets from one said process to other said process or by mapping new events to relatively idle resource sets.

94. A method in a computer apparatus for fault tolerant processing of at least one application in a plurality of processes running on at least one processor; the method comprising the steps of:

executing said application in a fault tolerant mode wherein said application is in an active condition on one process and is in standby condition on another said process on said processors;

representing said application by a single resource set;
and

an active single resource set modifying private data in said application and updating to a standby resource set of said application on another said process.

95. The method recited in claim 94, further comprising the steps of:

bringing said single resource set into either of active or standby state on said processes; and

said active resource set processing input events and sending update information to said standby resource set.

96. The method recited in claim 94, further comprising the step of using a warmstart procedure to bring said resource set into standby state from out of service state.

97. The method recited in claim 94, further comprising the step of providing communication between said application and other applications independent of application location and carrying out said communication external to the application by routing an event to the process where the resource set is active.

98. The method recited in claim 95, further comprising the step of transparently sending update messages from said active resource set to a corresponding said standby resource set by performing routing external to said application and routing messages to the process where the resource set is standby.

99. The method recited in claim 95, further comprising the steps of bringing said standby resource set into the active state for recovering from a failure of active resource set and routing events to new active resource set.

12209856.1